DS 705 Final Project

Jim Ryan

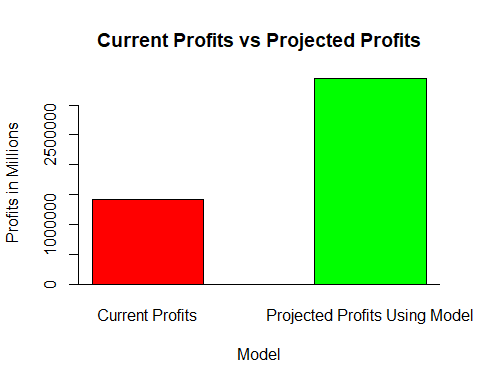
04/23/2020

Executive Summary

This report was commissioned to determine a way to predict which applicants are likely to default on their loans. This research provides a model that can be used to predict good loans from bad ones and in the process increase the bank’s profitability. The main method of analysis used was to use the results of past loans that the bank made to produce a model to predict loans that will be remain current (good loans) versus defaulted or charged off loans (bad loans) based upon 50,000 loans that the bank had made previously. The profitability based upon the current lending methods was determined along with the profitability of lending methods based upon the model created above. The model created above resulted in increased profit of $2,023,902 or an increase of 143%. The chart below shows he difference in profit between the current lending methods and the profit that would be realized had the model above been used. The model correctly predicts 80% of good loans. with the loans above that I analyzed only 78.1% of the loans were good loans

It is recommended that the lender implement the model above to increase profits, increase the number of good loans and decrease the number of bad loans.

The analysis has some limitations. The report could do a better job at predicting ‘bad’ loans. The most profitable model only correctly predicts 47% of bad loans. Another limitation of the analysis is the data itself. The type of lending institution is not known. nor is there a time frame as to when the data is collected. Depending on when the data was collected can have an effect on how well the analysis performs. Since 2007, there have been 2 major downturns in the economy resulting in rising employment rates and rising loan default rate. This is where the time frame in which the data would help. Another question that the data does not answer is what type of loans are in the dataset?. Are the loans secured,unsecured or both? Knowing this could certainly help the accuracy of the model. The data does not indicate over how long a period the data was collected. What is the monthly, quarterly and yearly loan figures. Is this data representative of one or many of those periods? This additional data would help the accuracy of this model.

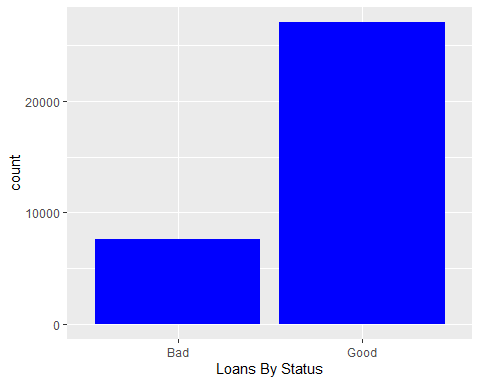


**Section 2: Introduction.**

For this project logistic regression will be used to predict which applicants are likely to default on their loans. The loans50k dataset that was given to us as part of this assignment was used. This data set contains 50,000 loans in various statuses and various amounts. There are 32 available variables in the data set. The response variable will be the status variable. Preparing and cleaning the data is first.

**Section 3: Preparing and Cleaning the Data**

The data contains different variables collected as either part of the application process or information from when the loan was disbursed. The goal is to predict which variable or variables will be able to predict whether a loan becomes default or not. All loans that did not have a status of ‘Fully Paid’, ‘Default’ or ‘Charged Off’ were removed. Then the response variable – status – was copied into another field called status2. Then the status of loans that were ‘Fully Paid’ were updated to ‘Good’ and the ones that were ‘Default’ or ‘Charged Off’ to ‘Bad’.

There are 27,074 loans or 78% with a status of ‘Good’ and 7,581 loans or 22 percent with a status of ‘Bad’. Here is a Histogram of that shows the number of good and bad loans after the status has been updated.  
 Sapply with a sum on the NA in the data set was used to determine which variable had NA in them and how many there were. Only 3 variables were found that had NA values in them. They were revolRatio (15 NA), bcOpen (360 NA) and bcRatio (384 NA).Since the missing values were only around 1 percent of the total number for these variables, they were imputed them using the mean of all the values for that specific variable.

Attention was focused on the employment variable. There were 15,268 different values for employment including 1918 where the value is blank. All rows where the value had a count greater than 100 were selected. 1789 values of the employment variable that had the word manager in it were updated to be just ‘Manager’. There were 934 different variations of the word ‘Teacher’, so all were updated to be ‘teacher’.

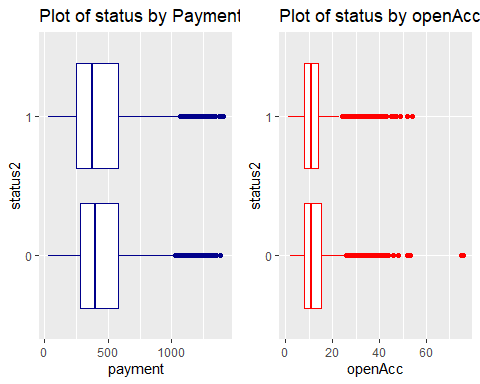
The ‘Grade variable was updated from ‘A’ thru ‘G’ to 0 thru 6 and made into a factor. The ‘verified’ variable was updated too. Those with a value of ‘Source Verified’ were up[dated to ‘Verified’ and that variable was made a factor. Something similar was done with the home variable. The Mortgage variable was set to OWN when the value was mortgage so then there were just 2 categorical variables - ‘own’ and ‘rent’ and the variable was also made a factor. The variable term only had 2 values (36 months and 60 months) so the variable term2 was updated to a factor.

The following variables were removed because more than 25 percent of them were 0: pubRec, delinq2yr, and inq6mth. The original status variable was also dropped.

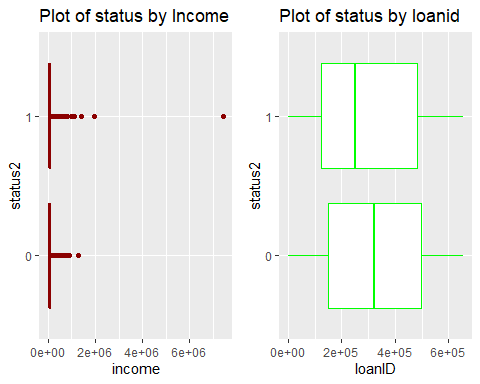
**Section 4: Exploring and Transforming the Data**

Now to delve into data exploration and transformation. First some exploration on the data set as a whole is done. Box plots of payment and status were created to see if there is an observable relationship between payment and status. There is no observable relationship between payment and status or the entire dataset. Then the same was done for openAcc income and loanID. The only one that showed an obvious relationship was totalBal. The ‘LoanID’ variable looks constant no matter what the status. This was assumed to be just a loan Identifier and was dropped. The other 3 variables were kept for Section 5.

bp1 <- ggplot(aes(x=payment,y=status2),data=LoansDT)+  
 geom\_boxplot(color='darkblue') + ggtitle("Plot of status by Payment")  
  
bp2 <- ggplot(aes(x=openAcc,y=status2),data=LoansDT)+  
 geom\_boxplot(color='red') + ggtitle("Plot of status by openAcc")  
  
bp3 <- ggplot(aes(x=income,y=status2),data=LoansDT)+  
 geom\_boxplot(color='darkred') + ggtitle("Plot of status by Income")  
  
bp4 <- ggplot(aes(x=loanID,y=status2),data=LoansDT)+  
 geom\_boxplot(color='green') + ggtitle("Plot of status by loanid")  
  
multiplot(bp1,bp2,cols=2)



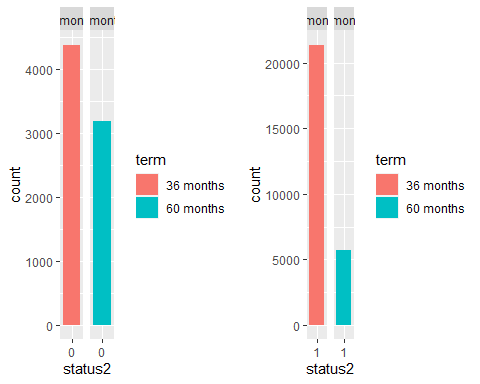
multiplot(bp3,bp4,cols=2)



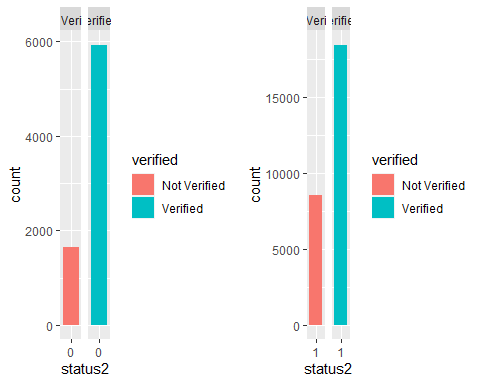
# LoansDT$loanID <- NULL

The modified data frame was divided into good and bad loans based on the status2 variable. Then bar graphs were made of some of the categorical variables divided by good and bad loans to see if there were any obvious relationships. The ‘term’, ‘verified’ and home variables wee plotted against status2 For the term variable. ‘Bad’ loans have a greater percentage of loans with 60 month terms than the ‘good’ loans. It appears that there might be some relationship between the ‘verified’ and term variables and status. This will be explored further for sectiont 5.

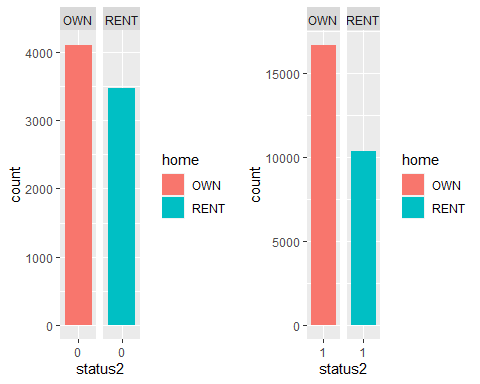
loans\_bad <- sqldf("Select \* from LoansDT where status2 = 0")  
loans\_good <- sqldf("Select \* from LoansDT where status2 = 1")  
  
   
p1 <- ggplot(loans\_bad, aes(status2, fill = term)) + facet\_wrap(~term) + geom\_bar()  
p2 <- ggplot(loans\_good, aes(status2, fill = term)) + facet\_wrap(~term) + geom\_bar()  
   
  
p3 <- ggplot(loans\_bad, aes(status2, fill = verified)) + facet\_wrap(~verified) + geom\_bar()  
p4 <- ggplot(loans\_good, aes(status2, fill = verified)) + facet\_wrap(~verified) + geom\_bar()  
   
  
p5 <- ggplot(loans\_bad, aes(status2, fill = home)) + facet\_wrap(~home) + geom\_bar()  
p6 <- ggplot(loans\_good, aes(status2, fill = home)) + facet\_wrap(~home) + geom\_bar()  
  
multiplot(p1,p2,cols=2)



multiplot(p3,p4,cols=2)



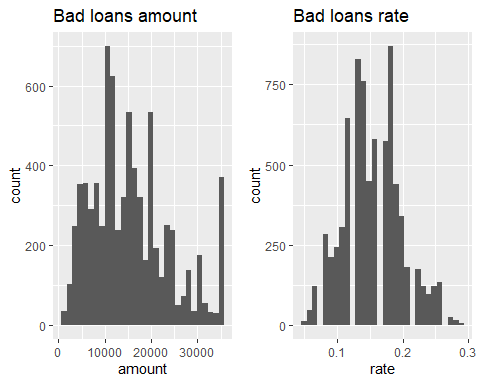
multiplot(p5,p6,cols=2)

 A histogram of all the variables in the LoansDT dataframe was plotted to get an overall sense of if the data is normally distributed or not. Due to space limitations, this was not included. Much of the data looks right skewed.

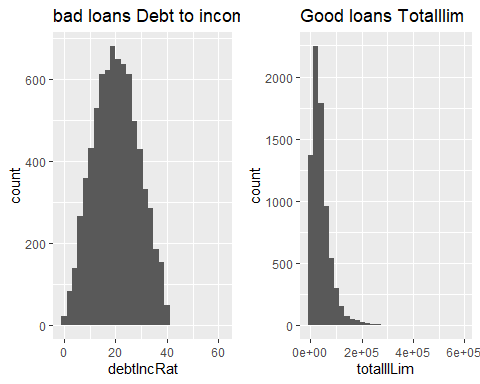
The states were plotted against status for both the good and bad loans but no relationship was observed. As expected the more populous states (New York and California) had the higher number of loans both good and bad. This variable will be dropped. The employment field is also going to be dropped as there are so many different values in there it will slow my regression down as there are over 15,000 distinct values, which is too many for the variable to be significant. Due to space concerns, the plots will not be displayed. Loanid will also be dropped from the dataset as that will only slow down the regression since the value for each loan is unique.

Variables from each dataframe (‘Good’ and ‘Bad’) were plotted to determine if the data is normally distributed or not. There is a definite right skew to many of the numeric variables. “rate”,“Amount”, “debt to income ratio” look normally distributed so they will not be transformed. To transform the rest of the numeric variables, 2 new dataframes for the good and bad loans were created and with just the numeric fields that were going to be transformed. Then the log1p function was used on the dataframes. This function was used instead of the log function because some of the variables contain zeroes and log has issues with zeroes. The following variables will not be transformed as they look normal: “revolRatio” , “accOpen24” and “accOpen24”.The data was plotted as a histogram again after the log and the data looks much more normally distributed. There are a few that now look left-skewed. Also after performing it the first time and looking for NA’s it was found that it created a total to 384 NAs in bcratio between the 2 groups so that variable was removed from the dataframes that log1p was applied to. The debtIncRat and rate fields were also normally distributed so they were not included in the group of variables to apply the log1p to. The variable “totalPaid” also was not transformed as it is not supposed to use that values as a predictor. The categorical variables were split into a separate dataframe to join with the logged dataframes later. Several of the variables were plotted after the log was applied to the to check the results.

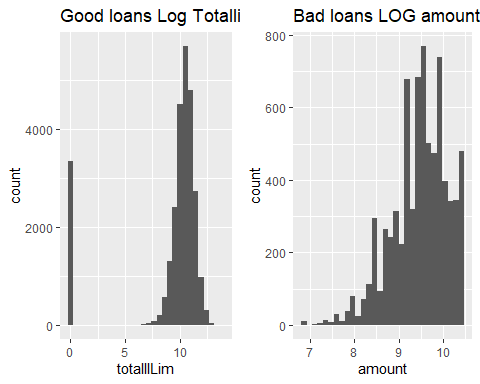
plt1 <- ggplot(loans\_bad, aes(x=amount)) + geom\_histogram() + ggtitle("Bad loans amount")  
plt2 <- ggplot(loans\_bad, aes(x=rate)) + geom\_histogram() + ggtitle("Bad loans rate")  
plt3 <- ggplot(loans\_bad, aes(x=debtIncRat)) + geom\_histogram() + ggtitle("bad loans Debt to income ratio")  
plt4 <- ggplot(loans\_bad, aes(x=totalIlLim)) + geom\_histogram() + ggtitle("Good loans TotalIlim")  
  
nologvars <- subset(LoansDT,select=c(bcRatio,debtIncRat,rate,totalPaid))  
  
good\_loan\_numeric <- loans\_good [ c("amount" , "payment" , "income" , "openAcc" , "revolRatio" , "totalAcc" ,   
 "totalBal" , "totalRevLim" , "accOpen24" , "avgBal" ,"bcOpen" , "totalLim",   
 "totalRevBal" , "totalBcLim" , "totalIlLim")]  
  
bad\_loan\_numeric <- loans\_bad[ c("amount" , "payment" , "income" , "openAcc" , "revolRatio" , "totalAcc" ,   
 "totalBal" , "totalRevLim" , "accOpen24" , "avgBal" ,"bcOpen" , "totalLim",   
 "totalRevBal" , "totalBcLim" , "totalIlLim")]  
  
good\_loan\_discrete <- loans\_good[ c("grade", "length" ,"home" ,"verified","reason" ,  
 "status2","term" )]  
  
bad\_loan\_discrete <- loans\_bad[ c("grade","length" ,"home" ,"verified","reason",   
 "status2","term")]  
  
  
good\_loan\_log <- log1p(good\_loan\_numeric)  
bad\_loan\_log <- log1p(bad\_loan\_numeric)  
  
plt5 <- ggplot(good\_loan\_log, aes(x=totalIlLim)) + geom\_histogram() + ggtitle("Good loans Log Totallim")  
plt6 <- ggplot(bad\_loan\_log, aes(x=amount)) + geom\_histogram() + ggtitle("Bad loans LOG amount")  
  
multiplot(plt1,plt2,cols=2)



multiplot(plt3,plt4,cols=2)



multiplot(plt5,plt6,cols=2)



The dataframes resulting from the log1p function was checked for NA’s and none was found. This is not shown to conserve space.

Next some density plots with some of the different logged variables were done to determine if there is a noticeable difference between good and bad loans. Density plots of openacc and totalLim for good and bad loans were done. Not much difference was observed for openacc or totalLim between good or bad loans.Not many fields were eliminated in the data exploration but once the model is built, that is when more fields will be eliminated.

p1 <- ggplot(good\_loan\_log, aes(x=openAcc)) +   
 geom\_density() + geom\_vline(aes(xintercept=mean(openAcc)),  
 color="blue", linetype="dashed", size=1)  
  
p2 <- ggplot(bad\_loan\_log, aes(x=openAcc)) +   
 geom\_density() + geom\_vline(aes(xintercept=mean(openAcc)),  
 color="green", linetype="dashed", size=1)  
  
  
p3 <- ggplot(good\_loan\_log, aes(x=totalLim)) +   
 geom\_density() + geom\_vline(aes(xintercept=mean(totalLim)),  
 color="red", linetype="dashed", size=1)  
  
p4 <- ggplot(bad\_loan\_log, aes(x=totalLim)) +   
 geom\_density() + geom\_vline(aes(xintercept=mean(totalLim)),  
 color="orange", linetype="dashed", size=1)

**PART 2 - Section 5:- The Logistic Model**

For the start of Section 5, 2 datasets were created from the Loan\_regrsn dataframe from Step 4. One dataset will be the training dataset and will contain 80% of the data, the other will be the test dataset and will contain 20% of the data. there were 6931 in Loan\_test and 27724 in Loan\_training.

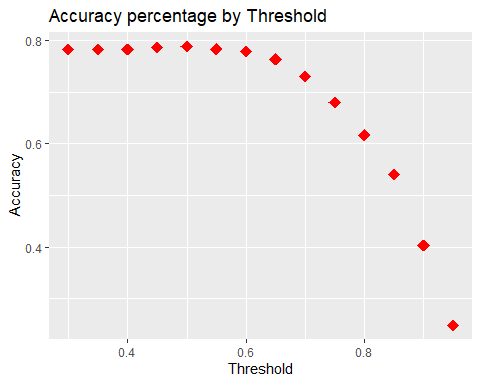
Next the full model is run along with the summary function to see which variables are significant and which should be included in the model. Based upon the coefficients with significant p values the following values are kept: grade,verified,reason,term,income,revolRatio,totalAcc,totalRevLim,accOpen24,bcOpen,totalRevBal,totalIlLim. When this model is run, the p-values for the following fields are no longer significant :bcOpen,totalRevBal and totalIlLim, so they are dropped then the regression is tried a third time. The McFadden pseudo R Squared for the third model (.0925) was less than the second model (.0929) but just barely.Looking at the model only 3 of the 12 values of the reason variable were significant so the reason variable was dropped and regression was run again. The results were much better for the 4th time. The p-value is very small (< 2e-16) with a McFadden Pseudo R2 of 0.092. This is the model to move forward with. To save space, only the p-values will be discussed and the output of the models will not be shown.

Below the model created above is used to predict the status for loans in the test data. A confusion matrix is created to determine the overall accuracy of the model. The results of the confusion matrix for the model are as follows: 5260 loans correctly predicted as good loans and 154 loans correctly predicted as bad loans for a total accuracy percentage of 78%. This matches with the baseline of 78% of the loans in the dataset are ‘Good’. 98% of good loans were correctly predicted as good while only 11% of bad loans were correctly predicted as bad. While this model leaves a bit to be desired in predicting the bad loans accurately, it does a much better job accurately predicting the good ones.

**Section 6 - Optimizing the Threshold for Accuracy.**

In this section the threshold is varied from .5 to attempt to correctly predict more bad loans. A contingency table is created for each different threshold and then graphed to show accuracy vs. threshold. As you recall from the previous section my accuracy percentage was 78% but was a much better predictor of ‘good’ loans than ‘bad’ loans.The accuracy will be calculated for 12 different thresholds between 0 and 1 to attempt to predict a better percentage of bad loans. Due to space constraints, the results will not be shown. The way that threshold accuracy calculation works is that probabilities are calculated using the model we chose above. The calculated probability is between 0 and 1. the Threshold value is a set number between between 0 and 1. Any calculated probabilities that are greater than or equal to that threshold are considered ‘Good’ while probabilities below the threshold are considered bad.

Given the data below, the accuracy vs the threshold of each of them is determined - including the default (.5). So a frequency table was built for 14 thresholds between .30 and .95. The highest accuracy level is at the .50 threshold. The accuracy percentages is 78.81% for the default threshold and 78.57 for the .45 threshold. The default threshold does a little better job at predicting ‘bad’ loans with 141 correctly predicted as ‘bad’ compared to 63 for the .45 threshold, but the .45 threshold does a better job at predicting ‘good’ loans with 5341 ‘bad’ loans predicted correctly versus 5260 for he default threshold. The lower the threshold is ,the better it is at predicting good loans. Unfortunately, the lower the threshold, the worse it is a predicting ‘bad’ loans. The lowest threshold I used, .3, is 78.26% accurate with 5415 out of 6931 loans correctly predicted as good but it incorrectly predicts all but 9 ‘bad’ loans as good. As the threshold increases from the default level, the overall accuracy drops as the number of ‘good’ loans predicted correctly drops but the number of bad’ loans predicted correctly increases



**Section 7 - Optimizing the Threshold for Profit.**

In this section the predictions at the different threshold levels from Part 6 will be taken and applied to the test data. Then the profit for each threshold will be summed to determine what the maximum profit increase is for the loans predicted as good. The probabilities calculations from section 5 and the test data created in section 5 are added . Then the total paid minus the amount is calculated for the total profit. According to the calculations, the threshold of .70 is where the biggest profit is $3,442,612.00. AT this threshold, it correctly predicted 80% of the good loans and 47% of the bad loans.

The lowest profit is the last 2 thresholds .90 of.95 where the profit is $1,420,765.00. The second highest profit among thresholds is at .65 where the profit is $3,279.712.00. The default threshold profit (.5) is the $2,285,,551.00. The other thresholds are as follows: .30 and .35 are the same at $1,519,487.00, .40 = $1,540,769.00, .45 = $1,848,694.00 ,.55 = $2,567,731.00, .60 = $3,043,274.00,.75 = 3,112,421.00, .80 = $3,112,421.00, and .85 = $2,459,591.00.

The total profit for all the loans in the test set is $1,418,710. Compared to not using any model the maximum percentage increase that can be expected by deploying the model is 143%. The profit for a ‘perfect’ model that denies all of the truly bad loans is $12,435,235. The increase in profit from the model above to the perfect model is $8,992,623 or 261%. For the best profit threshold, the overall accuracy of correctly predicted bad and good loans is 78.11%. This matches up with the percentage of good loans in the beginning dataset. The model above shows a profit increase of $2,023,902 or a profit increase of 142%. The maximum profit threshold does not coincide with the maximum accuracy threshold. The maximum accuracy threshold is the default threshold of .5 whereas the maximum profit threshold is at .70. The accuracy percentage at .5 is 79% while at the .7 threshold is 73%. As the threshold increases from .3 to .95, the number of accurately predicted good loans decreases where the number of bad loans accurately predicted increases. The best trade-off as far as profit goes is at the .7 threshold.

**Section 8 - Results Summary**

The model has a binary response (outcome, dependent) variable called status2. The values are ‘good’ or ‘bad’. The final classification model ended up with 8 predictor variables: grade, verified ,term, income, revolRatio, totalAcc, totalRevLim and accOpen24. The p-value of the final model is .0006. the AIC is 26350. It took me 4 different models to arrive at the final model. the final model results in a percent increase in profit, or $ $2,023,902 for the bank over their current process. If the bank used a perfect model and denied all the loans in the dataset that are marked as bad, it would result in an 2 percent increase in profit or 750,778.40 total profit. The final model had a threshold of .70 and correctly predicted 5,413 out of 6,931 loans correctly. That is 78.1%. the model correctly predicted 5,341 out of 6,931 or 99.3% ‘good loans’. It correctly predicted 72 out of 1,555 ‘bad’ loans’ or 5%. The final model is extremely accurate in predicting good loans.